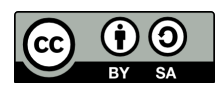
ECAM

Brussels Engineering School

Deploying VM with *Virsh* on a headless Linux server

Rémy Taymans

June 2018



Contents

Concepts

Tools

Demo

KVM

Kernel-based Virtual Machine

- Is a kernel module
- Does no emulation
- Uses modern extension of physical processor (AMD-V & Intel VT)

Type 1 hypervisor

Faster than emulating a hole processor

QEMU

Hosted virtual machine monitor

Operating modes:

- **User-mode emulation:** run single programs compiled for a different instruction set.
- **System emulation:** emulate a full computer system, including peripherals. (type 2 hypervisor)
- **KVM Hosting:** emulate a full computer system, but the execution of the guest is done by KVM.
- **Xen Hosting:** emulate a full computer system, but the execution of the guest is done by Xen.

Concepts

Tools

Demo

Libvirt (1)

API, deamon and management tool for managing platform virtualisation.

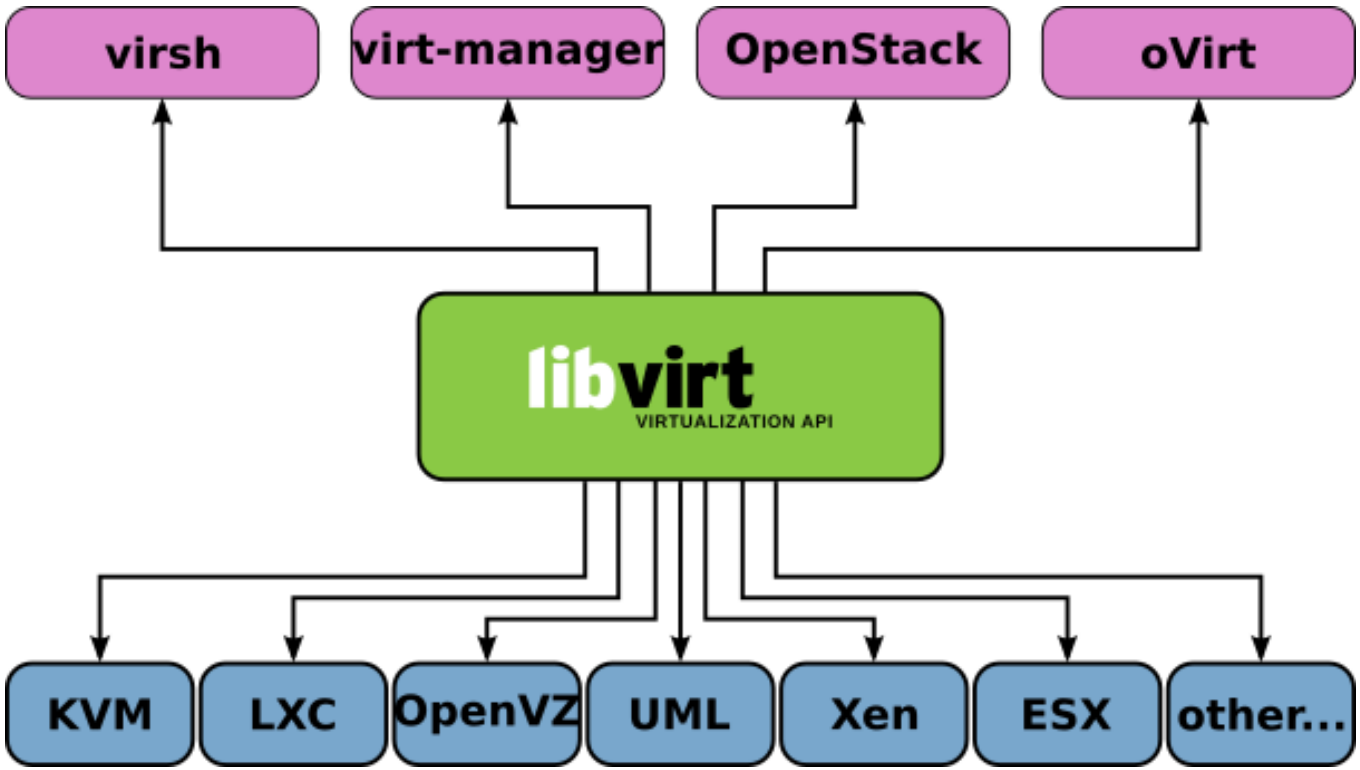


Figure 1: Wikimedia, Shmuel Csaba Otto Traian, CC-BY-SA, 2018

Libvirt (2)

CLI tools can be implemented using `libvirt`.

- `virt-install`
- `virt-builder`

But also Virsh

Virsh

CLI tool for managing virtual machine based on `libvirt`

- `virsh list`
- `virsh dumpxml <vm_name>`
- `virsh console <vm_name>`
- `virsh start <vm_name>`
- `virsh shutdown <vm_name>`
- `virsh destroy <vm_name>`

Concepts

Tools

Demo

Steps

1. Connect to remote server
2. Install needed packages
3. Create a bridge network interface
4. Create disk image
5. Install VM
6. Control and access the VM

Connect to remote server

With display:

```
ssh -X user@server.srv
```

Without display:

```
ssh user@server.srv
```

Installation

Debian package needed:

- **qemu-kvm:** QEMU.
- **libvirt-deamon-system:** Libvirt tools
- **virtinst:** set of CLI tools to create virtual machines using libvirt (e.g. `virt-install`, `virt-clone`, etc.)
- **libguestfs-tools:** set of CLI tools to manage guest disk images (e.g. `virt-builder`, `virt-resize`, etc.)
- **xtightvncviewer:** simple VNC viewer.
- **net-tools:** network tools (e.g. arp, netstat, noute, etc.)

User should be in groups `libvirt` and `libvirt-qemu`.

Create a bridge network interface (1)

Create a bridge interface `/etc/network/interfaces`:

```
auto br0
iface br0 inet dhcp
      bridge_ports enp0s3
```

Add bridge to QEMU with `/root/bridged.xml`:

```
<network>
  <name>br0</name>
  <forward mode="bridge"/>
  <bridge name="br0"/>
</network>
```

Create a bridge network interface (2)

Activate the bridge for VM:

```
# virsh net-define --file /root/bridged.xml
# virsh net-autostart br0
# virsh net-start br0
```

Create a disk image

List all the OS installable:

```
$ virt-builder --list
```

Get more info about an OS:

```
$ virt-builder --notes debian-9
```

Create the VM:

```
# virt-builder debian-9 \
    --size 6G \
    --format qcow2 -o /foo/bar/debian9-vm.qcow2 \
    --hostname debian9-vm \
    --network \
    --timezone Europe/Brussels
```

Install VM

Finally import the image with virt-install command:

```
# virt-install --import --name debian9 \
    --os-variant debian9 \
    --vcpus 1 \
    --memory 512 \
    --disk path=/foo/bar/debian9-vm.qcow2 \
    --network bridge=br0,model=virtio \
    --graphics vnc \
    --noautoconsole # do not open console
```

Control and access the VM

Using the console:

```
# virsh list
# virsh console --safe <my_vm_name>
```

Using VNC:

```
# virsh vncdisplay <my_vm_name>
localhost:0
# vncviewer localhost:0
```

Questions

Any questions ?